PATENT ABSTRACTS OF JAPAN

(11)Publication number : 02-027389
(43)Date of publication of application : 30.01.1990

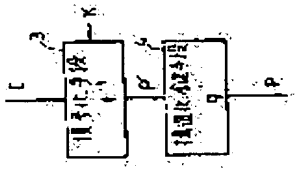
(51)Int.CI. G09C 1/00

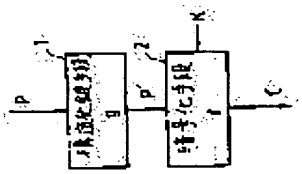
(21)Application number : 63-176775
(22)Date of filing : 15.07.1988

(71)Applicant : SONY CORP
(72)Inventor : NODA SHIGETOSHI

(54) ENCIPHERING METHOD AND ENCIPHERING DEVICE/DECODING DEVICE USING ENCIPHERING METHOD CONCERNED

(57)Abstract:
PURPOSE: To eliminate the danger of decoding by strengthening the cipher strength by only adding a circuit for executing a processing of a simple algorithm to strong enciphering/decoding circuits of a DES system, an FEAL system, etc.
CONSTITUTION: A structured key 1 makes a code p' from a plain sentence (p) from an algorithm (g), and (g) is a secret. An enciphering means 2 makes a cipher text (c) by executing strongly an encipherment by an algorithm (f) by using the code p' and a key (k). (f) can be disclosed. A decoding means 3 makes a code from the cipher text (c) by an algorithm f−1 by using (k), and a key means 4 decodes the plain sentence (p) from p'. In this case, conditions of f = f−1, random correspondence f(p', k) = Rand (p', k), f(,, p', ,, k) = Rand (,, p', ,, k), encipherment c = f(p'), and decoding p' = f−1(c) are satisfied. When p' and (c) are known, (k) can be decoded by a detailed inspection, but when one of them is unknown, decoding is impossible. (f) is obtained by using DES and FEAL systems. Also, g = g−1, and a bit transposition of a simple structure is used. According to this constitution, decoding becomes impossible even in the case of a high speed operation processing.





LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision

of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

⑲日本国特許庁(JP)　⑪特許出願公開

⑫公開特許公報(A)　平2−27389

⑤Int.Cl.⁵	識別記号	庁内整理番号	㊸公開　平成2年(1990)1月30日
G 09 C 1/00		7368−5B	

審査請求　未請求　請求項の数 6　(全10頁)

㊹発明の名称　暗号化方法及び該暗号方法を用いた暗号化装置／復号化装置

㉑特　願　昭63−176775

㉒出　願　昭63(1988)7月15日

㉒発明者　納田　重利　東京都品川区北品川6丁目7番35号　ソニー株式会社内

㉑出願人　ソニー株式会社　東京都品川区北品川6丁目7番35号

㉔代理人　弁理士　杉浦　正知

明　細　書

1.発明の名称

暗号化方法及び該暗号方法を用いた暗号化装置／復号化装置

2.特許請求の範囲

(1) 入力コードをx、キーコードをk、任意の入力コードのビット変化高をΔx、任意のキーコードのビット変化高をΔkとしたとき、

$f=f^{-1}$又はf^{-1}が存在し、

$f(x,k)=Rand(x,k)$

$f(\Delta x,\Delta k)=Rand(\Delta x,\Delta k)$

を満足するようなアルゴリズムfの暗号化を行う際の鍵として、上記アルゴリズムの鍵と独立で且つ鍵自体が暗号化アルゴリズムをなす構造化鍵を用いるようにした暗号化方法。

(2) 入力コードをx、キーコードをk、任意の入力コードのビット変化高をΔx、任意のキーコードのビット変化高をΔkとしたとき、

$f=f^{-1}$又はf^{-1}が存在し、

$f(x,k)=Rand(x,k)$

$f(\Delta x,\Delta k)=Rand(\Delta x,\Delta k)$

を満足するようなアルゴリズムfの暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした暗号化装置。

(3) 入力コードをx、キーコードをk、任意の入力コードのビット変化高をΔx、任意のキーコードのビット変化高をΔkとしたとき、

$f=f^{-1}$又はf^{-1}が存在し、

$f(x,k)=Rand(x,k)$

$f(\Delta x,\Delta k)=Rand(\Delta x,\Delta k)$

を満足するようなアルゴリズムfの暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした復号化装置。

(4) 入力コードをx、キーコードをk、任意の入力コードのビット変化高をΔx、任意のキーコードのビット変化高をΔkとしたとき、

$f=f^{-1}$

$f(x,k)=Rand(x,k)$

$f(\Delta x,\Delta k)=Rand(\Delta x,\Delta k)$

を満足するような暗号化アルゴリズムfの処理を

THIS PAGE BLANK (USPTO)

行う前後に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの処理を設けるようにした暗号化方法。

(5) 入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

$f = f^{-1}$

$f(x, k) = \mathrm{Rand}(x, k)$

$f(\Delta x, \Delta k) = \mathrm{Rand}(\Delta x, \Delta k)$

を満足するような暗号化アルゴリズムｆの暗号化回路の前段と後段に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの暗号化回路を設けるようにした暗号化装置。

(6) 入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

$f = f^{-1}$

$f(x, k) = \mathrm{Rand}(x, k)$

$f(\Delta x, \Delta k) = \mathrm{Rand}(\Delta x, \Delta k)$

を満足するような暗号化アルゴリズムの復号化回路の前段と後段に、少なくとも互いに逆関数となる構造を持つ暗号化アルゴリズムの復号化回路を設けるようにした復号化装置。

3.発明の詳細な説明

〔産業上の利用分野〕

この発明は、コンピュータネットワークシステムで通信されるデータの保護のために用いられる暗号化方法及びその暗号化装置／復号化装置に関する。

〔発明の概要〕

この発明は、コンピュータネットワークシステムで通信されるデータの保護のために用いられる暗号化方法及びその暗号化装置／復号化装置において、ＤＥＳ方式やＦＥＡＬ方式のように強い暗号化を行うアルゴリズムと直列に、鍵自体が暗号化アルゴリズムの構造化鍵を付加することにより、暗号化強度を強化するようにしたものである。

また、この発明は、コンピュータネットワークシステムで通信されるデータの保護のために用いられる暗号化方法及びその暗号化装置／復号化装置において、ＤＥＳ方式やＦＥＡＬ方式のように強い暗号化を行うアルゴリズムの前段と後段に、弱い暗号化アルゴリズムを付加することにより、暗号強度を強化するようにしたものである。

〔従来の技術〕

暗号化技術は、アタッカーにより情報が盗用或いは漏洩されることを防止するとともに、相互認証の基で確実に通信し合うことを可能にする。本格的に到来するであろう大規模分散情報通信ネットワークシステムにおいて、情報を保護していくために、このような暗号化技術の発達とその普及は不可欠である。

暗号方式には、大別して慣用鍵方式（共通鍵方式）と公開鍵方式とがある。慣用鍵方式では、暗号化鍵と復号化鍵とが共通とされる。公開鍵方式では、暗号化鍵と復号化鍵とが異なり、暗号化鍵が公開される。

暗号方式には、種々の方式が提案されている。その中で実用的な符号としては、慣用鍵方式の暗号化方式においては、ＤＥＳ（Data Encryption Standard）方式と、ＦＥＡＬ（Fast Data Encipherment Algolism）方式があり、公開鍵暗号化方式においては、ＲＳＡ(Rivest Shamir Adleman)方式がある。

ＤＥＳ方式やＦＥＡＬ方式は、強い暗号化を行うアルゴリズムにより暗号化が行われるため、虱み潰し以外に解読できないとされている。このような強い暗号化方式では、入力される平文と出力される暗号文及び鍵と出力される暗号文との関係がランダムに結ぶものと言える。このような関係は、入力される平文をｐ、鍵をＫとした時、

$f(p, K) = \mathrm{Rand}(p, K)$

と表せる。

更に、このような強い暗号化方式では、入力される平文のビット変化高に対する出力される暗号文のビット変化高及び鍵のビット変化高に対する出力される暗号文のビット変化高がランダムであり、

THIS PAGE BLANK (USPTO)

f（Δp，ΔK）＝Rand（Δp，ΔK）

であるように構成されている。

一般に、このような暗号化方式のアルゴリズムは、インボリューション構造を持つ。インボリューション構造は、

c＝f（p），p＝f⁻¹（c），

において

f＝f⁻¹

が成立する構造である。インボリューション構造を持つ場合には、暗号化と復号化が同様の処理プロセスで行える。ＥＸ－ＯＲをとるアルゴリズム（ｍｏｄ２の加算を行うアルゴリズム）は、インボリューション構造の簡単な例である。

〔発明が解決しようとする課題〕

暗号化方式としては、上述したように、種々のものが提案されている。そして、信頼性の保証とハードウェアの共通化をはかれるために、暗号化方式を標準化することが検討されている。

ところが、超並列処理コンピュータの開発等、コンピュータ技術の発展により、上述した従来の暗号化方式では、十分安全であるとは言えなくなってきている。このことが、暗号化方式を標準化していく上での一つの障害となっている。

つまり、暗号の強さのひとつのパロメータとして鍵のビット長がある。すなわち、鍵のビット長が長くなればなるほど、虱み潰しで解読される危険性が少なくなる。したがって、鍵のビット長を増加していくことで、暗号強度を増加できる。

ところが、鍵のビット数が増えると、処理が複雑化して、コストパフォーマンスが悪くなる。適当な妥協点として、ＤＥＳ方式やＦＥＡＬ方式では、従来、ビット長を６４ビットとしている。

このようにした場合、例えば、ＤＥＳ方式では、虱み潰しで暗号を解読するのに、2^{55}回の演算が必要である。ＦＥＡＬでは、虱み潰しで暗号を解読するのに、2^{64}回の演算が必要である。したがって、並列処理を行わず、１μｓで１回の速度で演算を行って解読する場合には、虱み潰しで解読を行うのに、ＤＥＳ方式では１１００年程かかり、ＦＥＡＬ方式では５１万年程かかることになり、ＤＥＳ方式やＦＥＡＬ方式は、十分安全な暗号といえる。

ところが、近年のコンピータ処理速度の向上は目覚ましく、10^6の並列処理で演算を行うことが実現可能になった場合には、虱み潰しで解読を行うのに、ＤＥＳ方式では９．６時間程、６．１カ月程で良いことになる。

したがって、この発明は、このような超並列型のコンピュータの開発等、将来の演算速度の向上に備えて、より安全な暗号化を行える暗号化装置の提供を目的とする。

〔課題を解決するための手段〕

この発明は、入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

f＝f⁻¹又はf⁻¹が存在し、

f（ｘ，ｋ）＝Rand（ｘ，ｋ）

f（Δｘ，Δｋ）＝Rand（Δｘ，Δｋ）

を満足するようなアルゴリズムｆの暗号化を行う際の鍵として、上記アルゴリズムの鍵と独立で且つ鍵自体が暗号化アルゴリズムをなす構造化鍵を用いるようにした暗号化方法である。

また、入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

f＝f⁻¹又はf⁻¹が存在し、

f（ｘ，ｋ）＝Rand（ｘ，ｋ）

f（Δｘ，Δｋ）＝Rand（Δｘ，Δｋ）

を満足するようなアルゴリズムｆの暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした暗号化装置である。

また、入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

f＝f⁻¹又はf⁻¹が存在し、

f（ｘ，ｋ）＝Rand（ｘ，ｋ）

f（Δｘ，Δｋ）＝Rand（Δｘ，Δｋ）

THIS PAGE BLANK (USPTO)

を満足するようなアルゴリズムｆの暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした復号化装置である。

また、入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

$f = f^{-1}$

$f(x, k) = \mathrm{Rand}(x, k)$

$f(\Delta x, \Delta k) = \mathrm{Rand}(\Delta x, \Delta k)$

を満足するような暗号化アルゴリズムｆの処理を行う前後に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの処理を設けるようにした暗号化方法である。

また、入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

$f = f^{-1}$

$f(x, k) = \mathrm{Rand}(x, k)$

$f(\Delta x, \Delta k) = \mathrm{Rand}(\Delta x, \Delta k)$

を満足するような暗号化アルゴリズムｆの暗号化回路の前段と後段に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの暗号化回路を設けるようにした暗号化装置である。

また、入力コードをｘ、キーコードをｋ、任意の入力コードのビット変化高をΔｘ、任意のキーコードのビット変化高をΔｋとしたとき、

$f = f^{-1}$

$f(x, k) = \mathrm{Rand}(x, k)$

$f(\Delta x, \Delta k) = \mathrm{Rand}(\Delta x, \Delta k)$

を満足するような暗号化アルゴリズムの復号化回路の前段と後段に、少なくとも互いに逆関数となる構造を持つ暗号化アルゴリズムの復号化回路を設けるようにした復号化装置である。

〔作用〕

ＤＥＳ方式やＦＥＡＬ方式のように強い暗号化を行うアルゴリズムと直列に、鍵自体が暗号化アルゴリズムの構造化鍵を付加することにより、暗号化強度が強化される。

また、ＤＥＳ方式やＦＥＡＬ方式のように強い暗号化を行うアルゴリズムの前段と後段に、弱い暗号化アルゴリズムを付加することにより、暗号強度が強化される。

〔実施例〕

この発明の実施例について、以下の順序にしたがって説明する。

ａ．構造化鍵を用いて暗号強度を強化させる例

ａ１．構造化鍵について

ａ２．構造化鍵としてＥＸ－ＯＲを用いた場合の考察

ａ３．構造化鍵としてビット転置器を用いた場合の考察

ａ４．構造化鍵を用いた場合の実施例

ｂ．強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例

ｂ１．強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例の基本構成

ｂ２．強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる構成の実施例

ａ．構造化鍵を用いて暗号強度を強化させる例

ａ１．構造化鍵について

現代暗号規約において、暗号に対する信用性や安全性をはかるために、以下のような規約が提案されている。

①　鍵以外の暗号化処理が公開されている。

②　解読者は、鍵以外の全ての情報を知っているとして、鍵の虱み潰し解読法又は計算量的安全な方法以外に解読法が知られていない。

このような規約を遵守した上で、より強い暗号化を行う方法を考察することにする。

従来の暗号化アルゴリズムでは、鍵が単純なコードだけである。より強い暗号化を行うために、これに加えて構造化された鍵を用いることが考えられる。

THIS PAGE BLANK (USPTO)

すなわち、ＤＥＳ方式やＦＥＡＬ方式のように強い暗号化アルゴリズムと直列に、鍵自体が暗号化を行う構造の鍵を設けることが考えられる。このように、構造化された鍵を用いれば、解読がより困難になると考えられる。

このような構造化鍵は秘密にされる。そして、構造化鍵以外の全ての暗号化処理は、現代暗号化規約に則って公開される。構造化鍵は、記録媒体に記憶させ、秘密とされる関係上、単純な構造であることが望ましい。

第１図は、このような構造化鍵を用いた暗号化の処理を示すものである。第１図において、１は構造化鍵手段、２は暗号化手段である。構造化鍵手段１と暗号化手段２とが直列に配置される。

構造化鍵手段１は、アルゴリズムｇにより、入力される平文ｐを暗号化し、中間コードｐ´を生成するものである。この構造化鍵１でどのようなアルゴリズムで暗号化がなされているかは、秘密とされる。

暗号化手段２には、構造化鍵手段１から出力される中間コードｐ´が与えられるとともに、鍵Ｋが与えられる。暗号化手段２は、中間コードｐ´を、鍵Ｋを用いて、アルゴリズムｆにより強い暗号化を行い、暗号文ｃを生成するものである。この暗号化手段２のアルゴリズムは、公開することができる。

第２図は、このような構造化鍵を用いて暗号化された暗号文を復号する復号化の処理を示すものである。復号化は、第１図に示す暗号化の処理に対応している。

第２図において、３は復号化手段、４は構造化鍵手段である。復号化手段３は、鍵Ｋを用いて、アルゴリズムf^{-1}により暗号文ｃから中間コードｐ´を生成するものである。構造化鍵手段４には、復号化手段３で復号された中間コードｐ´が与えられる。構造化鍵手段４は、中間コードｐ´から平文ｐを生成するものである。これにより、暗号文ｃから平文ｐが解読される。

ここで、アルゴリズムｆ及びアルゴリズムｇについて説明する。アルゴリズムｆは、強い暗号化を行うものである。このアルゴリズムｆは、以下のような条件を満たしている。

①　$f = f^{-1}$（インボリューション構造）又はf^{-1}が存在する。

②　入力ｐ´、鍵Ｋとすると、

$$f(p', K) = \mathrm{Rand}(p', K)$$

つまり、ｐ´やＫに対して出力ｆ（ｐ´，Ｋ）がランダムに対応する。

③　入力のビット変化量をΔｐ´、鍵のビット変化量をΔＫとすると、、

$$f(\Delta p', \Delta K) = \mathrm{Rand}(\Delta p', \Delta K)$$

つまり、Δｐ´やΔＫに対して、出力変化量もランダムに対応する。

④、暗号化は、

$$c = f(p')$$

で表せ、復号化は、

$$p' = f^{-1}(c)$$

で表せる。但し、インボリューション構造ならば、

$$p' = f(c)$$ である。

⑤　入力ｐ´や出力ｃが知られている時、鍵Ｋの虱み潰し検査により鍵Ｋは解読されるが、ｐ´又はｃの一方が未知の時、その一方と鍵Ｋは解読不可能な構造をとる。

このようなアルゴリズムｆとしては、ＤＥＳ方式やＦＥＡＬ方式を用いることができる。

構造化鍵手段１で行うアルゴリズムｇは、以下のような構造を持つものである。

①　$g = g^{-1}$又はg^{-1}が存在する。

②、入力ｐのビット変化高Δｐに対して、

$$g(\Delta p) \neq \mathrm{Rand}(\Delta p)$$

であっても良い。

③、入力ｐと出力ｐ〜の両方が知られた時、ｇの構造が知られても良いが、ｇの構造やｇ内の変数が知られない時、ｐとｐ´の一方が知られても、他方を知ることができない。

アルゴリズムｇとしては、極めて単純な構造、例えば、ＥＸ－ＯＲ構造、ビット転置、換字表等が考えられる。後述するように、構造化鍵として用いるアルゴリズムｇとしては、ビット転置が好ましく、ＥＸ－ＯＲ構造では、暗号化を十分強化

THIS PAGE BLANK (USPTO)

することはできない。

a 2．構造化鍵としてＥＸ－ＯＲを用いた場合の考察

強い暗号化を行う暗号化処理に対する構造化鍵として、全くランダムに選ばれた鍵ｋによりＥＸ－ＯＲをとるアルゴリズムを採用した場合について考察する。

第３図は、構造化鍵のアルゴリズムｇとして、ＥＸ－ＯＲをとる構造とした場合の例を示すものである。第３図において、１１は構造化鍵手段であり、この場合には、この構造化鍵手段１１は、鍵ｋと入力ｐとのＥＸ－ＯＲをとる回路である。暗号化処理手段１２は、ＤＥＳ方式やＦＥＡＬ方式のような強い暗号化を行う暗号化回路である。

このように、ＥＸ－ＯＲをとるアルゴリズムを、強い暗号化方法のアルゴリズムに対する構造化鍵とした場合の暗号強度について考察していくことにする。なお、便宜上ｆ＝ｆ$^{-1}$とみなす。

このような暗号化は、平文をｐ、ＥＸ－ＯＲ回路から出力される中間コードをｐ′、暗号文をｃとすると、

$c = f(p', K),\ p' = p \oplus k$

で示される。また、復号化は、

$p = k \oplus p',\ p' = f(c, k)$

であり、ここで、ｆはランダム関数である。上式をまとめると、暗号化は、

$c = f(p \oplus k, K) = Rand(p \oplus k, K)$

と表せ、復号化は、

$p = k \oplus f(c, K) = k \oplus Rand(c, K)$

と表せる。

上式から、以下のことがわかる。

すなわち、暗号化手続きは、２つの鍵ｋとＫとのランダム関数によるものの、復号化手続きは、ひとつの鍵Ｋのランダム関数によるもののため、復号化の方向では、解読において、Ｋの虱み潰しは必要だが、ｋの虱み潰しを行う必要はない。

つまり、復号化の方向においては、

$f(c, K) + p = k = 一定$

という関係が成立する。このため、アタッカーは、既知なるｉ個のサンプル（c_i，p_i）を用いて、第４図に示すように、アルゴリズムｆの処理を行う回路１３とＥＸ－ＯＲをとる回路１４とからなる構造の処理回路によって鍵Ｋを検索できる。すなわち、Ｋの虱み潰しで、全てのサンプルｉ個において出力が一定となった時、その時の出力（ｋ）とＫが鍵として知られる。したがって、この手法の解読手数は、$i \cdot 2^B$である（Ｂ：鍵及び入出力コードのビット数）。

関数ｆは、ランダム関数であるため、出力（ｋ）が１回一定となった時、その時の出力（ｋ）と鍵（Ｋ）が本物でない確率（誤り率）は、コード数2^B個内のひとつのコードをとる確率2^{-B}に同等な程低い。つまり、ｉ＝２でも十分解読可能と言える。

以上のような考察結果から、構造化鍵として、ＥＸ－ＯＲをとるような構成では、十分に暗号強度を許可することにはならない。

a 3．構造化鍵としてビット転置器を用いた場合の考察

構造化鍵としてビット転置器を採用した場合について考察する。また便宜上ｆ＝ｆ$^{-1}$とするものとする。

ビット転置器は、第５図に示すように、入力コードの各ビットをランダムに入れ替えるものである。ビット転置器のアルゴリズムｇ（ｐ）は、

$g(p) = Rand(p)$

$g(\Delta p) \neq Rand(\Delta p)$

$g \neq g^{-1}$

として表せる。

第６図は、構造化鍵としてビット転置器を用いた場合の暗号化処理を示し、第７図は、その復号化処理を示している。

第６図に示す暗号化処理において、２１は構造化鍵手段としてのビット転置器である。ビット転置器２１により、アルゴリズムｇにより、入力のビットが転置される。２２は暗号化手段である。暗号化手段２２は、アルゴリズムｆにより、強い暗号化を行う。このアルゴリズムとしては、ＤＥ

THIS PAGE BLANK (USPTO)

S方式やＦＥＡＬ方式が用いられる。

また、第７図に示す復号化処理において、２３は復号化手段であり、復号化手段２３でアルゴリズムｆにより、暗号文ｃから中間コードｐ´が生成される。この中間コードｐ´が逆ビット転置器２４に供給され、逆ビット転置器２４でアルゴリズムg^{-1}により、逆ビット変換される。

この場合の暗号化は、暗号化手段２２のアルゴリズムをｆ、ビット転置器２１のアルゴリズムをｇ、入力される平文をｐ、中間コードをｐ´、暗号文をｃとすると、

$c = f(p', k) = f(g(p), K)$

として表せる。また、復号化は、

$p = g^{-1}(p') = g^{-1}(f(c, K))$

として表せる。

ｆで示されるアルゴリズムは、強い暗号化を行っているので、入力変数のビット変化高（Δｐ やΔＫ及びΔｃ）に対して、出力がランダムに対応ずけられるため、ビット変化高に対する解読攻撃に対しては、アルゴリズムｆの鍵Ｋを虱み潰しにより求め、アルゴリズムｇの結線を虱み潰しにより求める以外に解読できない。また、アルゴリズムｇは、その入力と出力に特別な関係が存在しないので、結局、ビット変化高の攻撃以外の攻撃に対しても、アルゴリズムｆとアルゴリズムｇの両方の検査以外に解読法はないと言える。

入力及び出力のビット数をＢとすると、アルゴリズムｆの鍵Ｋを虱み潰しで捜す場合、2^B回の演算が必要であり、アルゴリズムｇの結線の虱み潰し回数はＢ！であるから、このようなアルゴリズムで暗号化を行った場合、虱み潰しで解読するのに、両方で$2^B \cdot B!$回の演算が必要になる。Ｂ＝６４ビットとすると、

$B! \fallingdotseq 10^{89}$

$2^B \cdot B! \fallingdotseq 10^{108}$

となる。この場合には、１回の処理速度を１μｓとして、10^6の並列処理を行ったとしても、虱み潰しで解読を行うのに、約10^{89}年必要になる。

ａ４．構造化鍵を用いた場合の実施例

第８図は、このように構造化鍵を用いて暗号強度を強化して通信を行う実施例を示すものである。

第８図において、通信を行うコンピュータシステム３１には、構造化鍵手段３２及びその構造化鍵手段３２を管理するための鍵管理と、暗号／復号器の鍵管理を行う鍵管理手段３３が設けられている。この構造化鍵手段３２及び鍵管理手段３３は、例えばソフトウェアで処理される。暗号／復号器３５及びモデム３６は、コントロールライン３７を介してコンピュータシステム３１と結ばれている。

コンピュータシステム３１から回線３４を介してデータを出力する場合には、コンピュータシステム３１からのデータがこの構造化鍵手段３２のアルゴリズムｇにより、中間コードに変換される。この中間コードが暗号／復号器３５に供給される。そして、暗号／復号器３５により、アルゴリズムｆにより暗号化され、この暗号化されたデータがモデム３６を介して回線３４に出力される。

回線３４を介して伝えられてきたデータをコンピュータシステム３１で受信する場合には、回線３４を介して伝えられてきたデータがモデム３６を介して暗号／復号器３５に供給される。暗号／復号器３５で、アルゴリズムｆにより送られてきたデータが復号され、中間コードが生成される。

この中間コードがコンピュータシステム３１の構造化鍵手段３２に供給される。構造化鍵手段３２で中間コードからデータが復号される。

ｂ．強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例

ｂ１．強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例の基本構成

第９図は、ＤＥＳ方式やＦＥＡＬ方式のような強い暗号化を行う暗号化手段の前段と後段に弱い暗号化を行う暗号化手段を配することにより、暗号強度を強化するようにした例である。なお、以下の説明では、$f = f^{-1}$とするものとする。第９図において暗号化手段５１には、平文ｐが与えら

THIS PAGE BLANK (USPTO)

れる。暗号化手段３１は、アルゴリズムｈにより、弱い暗号化を行う。この暗号化手段５１には、鍵ｋが与えられる。暗号化手段５１からは、中間コードｐ´が出力される。

暗号化手段５１で生成された中間コードｐ´が強い暗号化手段５２に与えられる。強い暗号化手段５２は、アルゴリズムｆにより、強い暗号化を行う。強い暗号化手段５２には、鍵ｋとは独立な鍵Ｋが与えられる。暗号化手段５２から、暗号ｃが出力される。

強い暗号化手段５２で暗号化されたコードは、更に、暗号化手段５３に送られる。暗号化手段５３は、アルゴリズムh^{-1}により、弱い暗号化を行う。暗号化手段５３からは、暗号ｃ´が出力される。なお、アルゴリズムｈはインボリューション構造で、$h=h^{-1}$である。

第１０図は、その復号化処理を示している。第１０図において、暗号文ｃ´が復号化手段６１に与えられる。復号化手段６１で、アルゴリズムｈにより、暗号文ｃ´が復号され、暗号文ｃが復号化手段６１から出力される。

この暗号文ｃが強い復号化手段６２に与えられる。強い復号化手段６２は、アルゴリズムｆにより、復号を行うものである。復号化手段６２からは、中間コードｐ´が出力される。この中間コードｐ´が復号化手段６３に与えられる。

ここで、アルゴリズムｆは、以下のような条件を満たしている。

① $f=f^{-1}$（インボリューション構造）

② 入力ｐ´、鍵ｋとすると、

$f(p', k)=\mathrm{Rand}(p', k)$

つまり、ｐ´やｋに対して出力$f(p', k)$がランダムに対応する。

③ 入力のビット変化量を$\Delta p'$、鍵のビット変化量をΔkとすると、、

$f(\Delta p', \Delta k)=\mathrm{Rand}(\Delta p', \Delta k)$

つまり、$\Delta p'$やΔkに対して、出力変化量もランダムに対応する。

④ 暗号化は、

$c=f(p')$

で表せ、復号化は、

$p'=f(c)$である。

⑤ 入力ｐ´や出力ｃが知られている時、鍵Ｋの虱み潰し検査により鍵Ｋは解読されるが、ｐ´又はｃの一方が未知の時、その一方と鍵Ｋは解読不可能な構造をとる。

アルゴリズムｆとしては、ＤＥＳ方式やＦＥＡＬ方式を用いることができる。

また、アルゴリズムｈは、以下のような構造を持つものである。

① $h=h^{-1}$又はh^{-1}が存在する。

② 入力ｐのビット変化高Δpに対して、

$h(p, k)\neq\mathrm{Rand}(p, K)$

$h(\Delta p, \Delta k)\neq\mathrm{Rand}(\Delta p, \Delta K)$

であっても良い。

③ 入力ｐと出力ｐ´の両方が知られた時、ｇの構造が知られても良いが、ｇの構造やｇ内の変数が知られない時、ｐとｐ´の一方が知られても、他方を知ることができない。

このような構造とした場合の暗号化は、

$c=h^{-1}(f(h(p)))$であり、

復号化は、

$p=h^{-1}(f(h(p)))$である。つまり、インボリューション構造をなす。

ここで、$f(h(p))$に注目してみると、

$f(h(p))=f(p, K, k)$

つまり、独立変数ｐ，Ｋ，ｋの関数であり、然も、

$f(p, K, k)=\mathrm{Rand}(p, K, k)$

$f(\Delta p, \Delta K, \Delta k)=\mathrm{Rand}(\Delta p, \Delta K, \Delta k)$

が成立している。

つまり、この構造では、Ｋやｋの虱み潰し検査以外の解読は不可能である。Ｋやｋのビット長をＢとすると、この方法の解読回数は、虱み潰しで行った場合、2^{2B}回となる。すなわち、Ｂ＝６４とした場合には、2^{128}回の演算が必要になり、１μｓの演算速度で、10^{6}の並列処理を行ったとしても、平均約10^{18}年処理時間がかかることになり、十分安全な暗号と言える。

このようなアルゴリズムで符号化を行う場合の

THIS PAGE BLANK (USPTO)

アルゴリズムｈとしては、例えばＥＸ－ＯＲを用いることができる。すなわち、第１１図に示すように、このようなアルゴリズムの符号化は、ＥＸ－ＯＲ回路７１及び７２と、アルゴリズムｆの暗号化をし行う暗号化手段７２とにより構成できる。

ｂ２．強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる構成の実施例

第１２図は、このように強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させて通信を行う場合の実施例を示すものである。

第１２図において、通信を行うコンピュータシステム８１には、鍵（Ｋ及びｋ）８２及びその鍵３２を管理するための鍵管理手段８３が設けられている。この鍵８２は、鍵管理手段８３により、例えばソフトウェアで管理されている。

暗号／復号器８５及びモデム８６は、コントロールライン８７を介してコンピュータシステム８１と結ばれている。

コンピュータシステム８１から回線８４を介してデータを出力する場合には、コンピュータシステム８１からのデータが暗号／復号器８５に供給される。そして、暗号／復号器８５により、アルゴリズムｈにより弱い暗号化がなされた後、アルゴリズムｆにより強い暗号化なされ、更に、アルゴリズムh^{-1}により、弱い暗号化がなされ、この暗号化されたデータがモデム８６を介して回線８４に出力される。

回線８４を介して伝えられてきたデータをコンピュータシステム８１で受信する場合には、回線８４を介して伝えられてきたデータがモデム８６を介して暗号／復号器８５に供給される。暗号／復号器８５で、アルゴリズムｈによる復号がなされ、更に、アルゴリズムｆによる復号がなされ、更に、アルゴリズムh^{-1}による復号がなされ、送られてきたデータが解読される。

この解読されたデータがコンピュータシステム８１に供給される。

〔発明の効果〕

この発明によれば、ＤＥＳ方式やＦＥＡＬ方式等の強い暗号化／復号化回路に、簡単なアルゴリズムの処理を行う回路を付加するだけで、暗号強度を強化でき、並列処理コンピュータにより演算速度が向上された場合でも、解読の危険性がなくなり、データの安全性を高めることができる。

4.図面の簡単な説明

第１図は構造化鍵を用いた暗号化の説明に用いるブロック図．第２図は構造化鍵を用いた暗号の復号化の説明に用いるブロック図．第３図は構造化鍵としてＥＸ－ＯＲを用いた場合の説明に用いるブロック図．第４図は構造化鍵としてＥＸ－ＯＲを用いた場合の解読の説明に用いるブロック図．第５図はビット転置器の説明に用いる略線図．第６図は構造化鍵としてビット転置器を用いた場合の暗号化処理の説明に用いるブロック図．第７図は構造化鍵としてビット転置器を用いた場合の復号化処理の説明に用いるブロック図．第８図は構造化鍵を用いた場合の実施例のブロック図．第９図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした暗号化の説明に用いるブロック図．第１０図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした暗号の復号化の説明に用いるブロック図．第１１図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした暗号化において弱い暗号化処理にＥＸ－ＯＲを用いるようにした例の説明に用いるブロック図．第１２図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした場合の実施例のブロック図である。

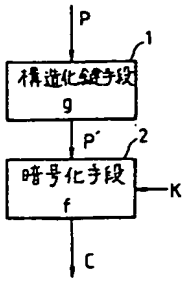
図面における主要な符号の説明

３１，８１：コンピュータシステム，

３２：構造化鍵手段．３５、８５：暗号／復号器。

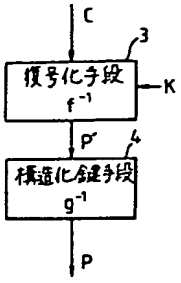
代理人　　弁理士　杉　浦　正　知

THIS PAGE BLANK (USPTO)



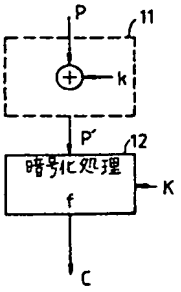
構造化鍵を用いた暗号化

第 1 図



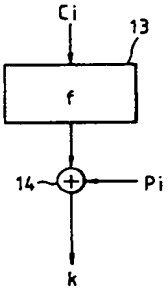
構造化鍵を用いた復号化

第 2 図



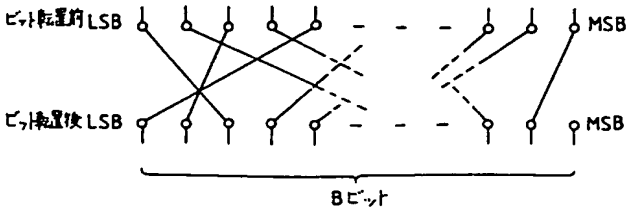
構造化鍵としてEX-ORを用いた場合の説明

第 3 図



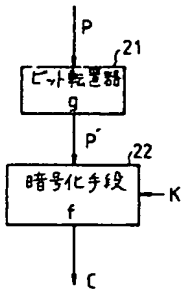
構造化鍵としてEX-ORを用いた場合の解読の説明

第 4 図



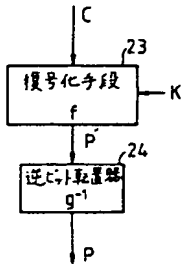
ビット転置器の処理

第 5 図



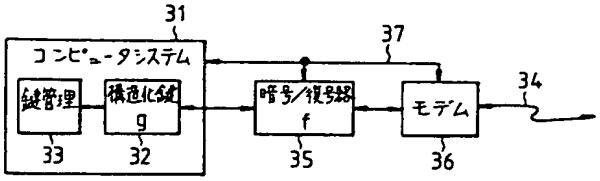
ビット転置器を用いた場合の暗号化処理

第 6 図



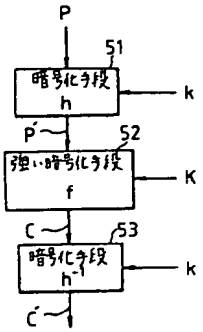
ビット転置器を用いた場合の復号化処理

第 7 図



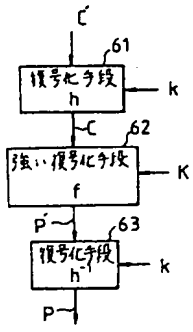
構造化鍵を用いた場合の実施例

第 8 図



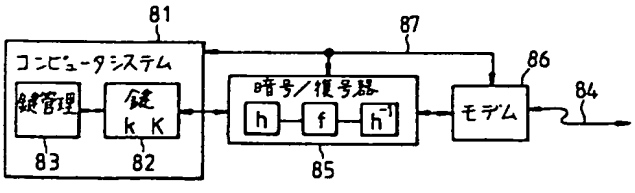
強い暗号化の前後に弱い暗号化を配する場合の暗号化の例

第 9 図



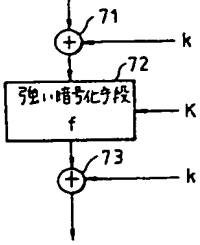
強い暗号化の前後に弱い暗号化を配する場合の復号化の例

第10図



強い暗号化の前後に弱い暗号化を配した場合の実施例

第12図



強い暗号化手段の前後にEX-ORを配した例

第11図

THIS PAGE BLANK (USPTO)